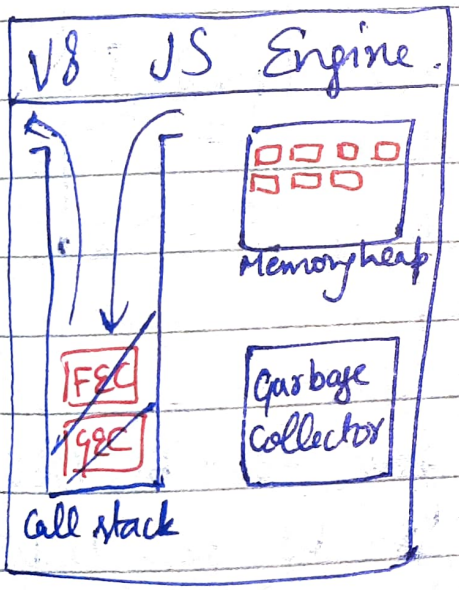
# EP-6 Asynchronous IO & Event Driven Architecture

JS → Synchronous (line by line)
→ Single threaded.

There is only synchronous execution available for JS.

JS is sync. until NODE JS comes. hehe

→ How Synchronous code executes??




```
Var a = 123;
Var b = 345;
function mul (x,y){
  const res = a * b;
  return res;
}
Var c = mul (a,b);
```

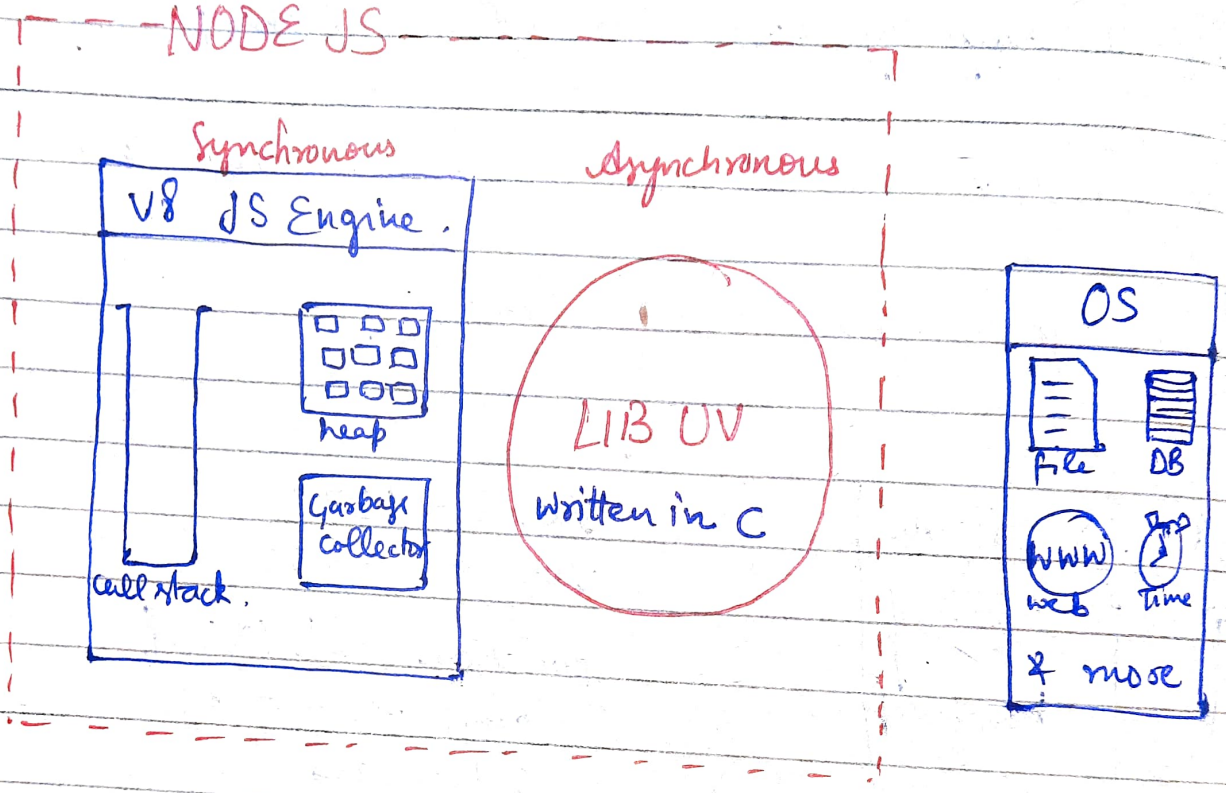
Step: 1 line by line stores var & funcs in heap.
Step: 2 stores a Global execution context in callstack.
Step: 3 if finds a func. call, it creates functional execution context.

# → How Async JS code works ??

There is no way a async code can work in JS,

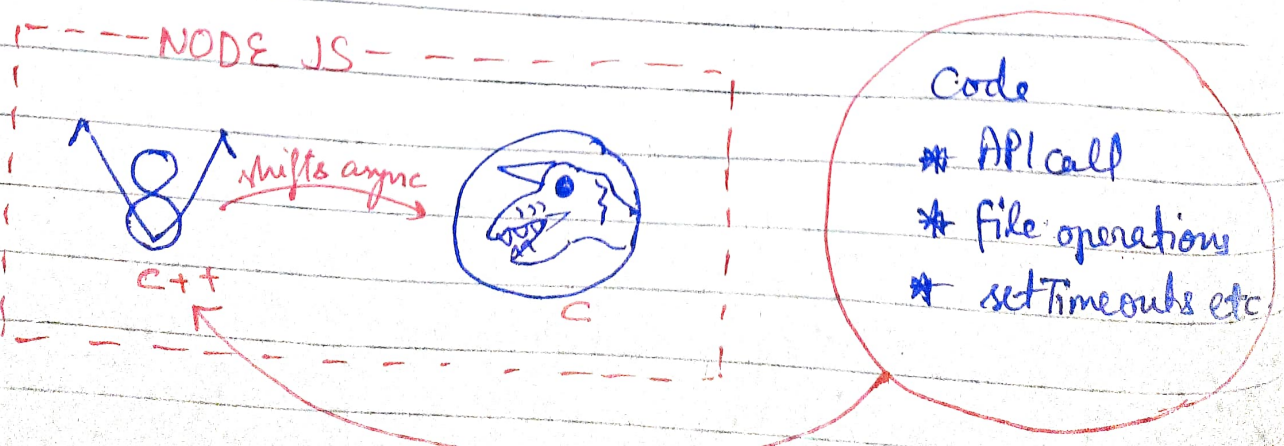
here comes NODE JS :→

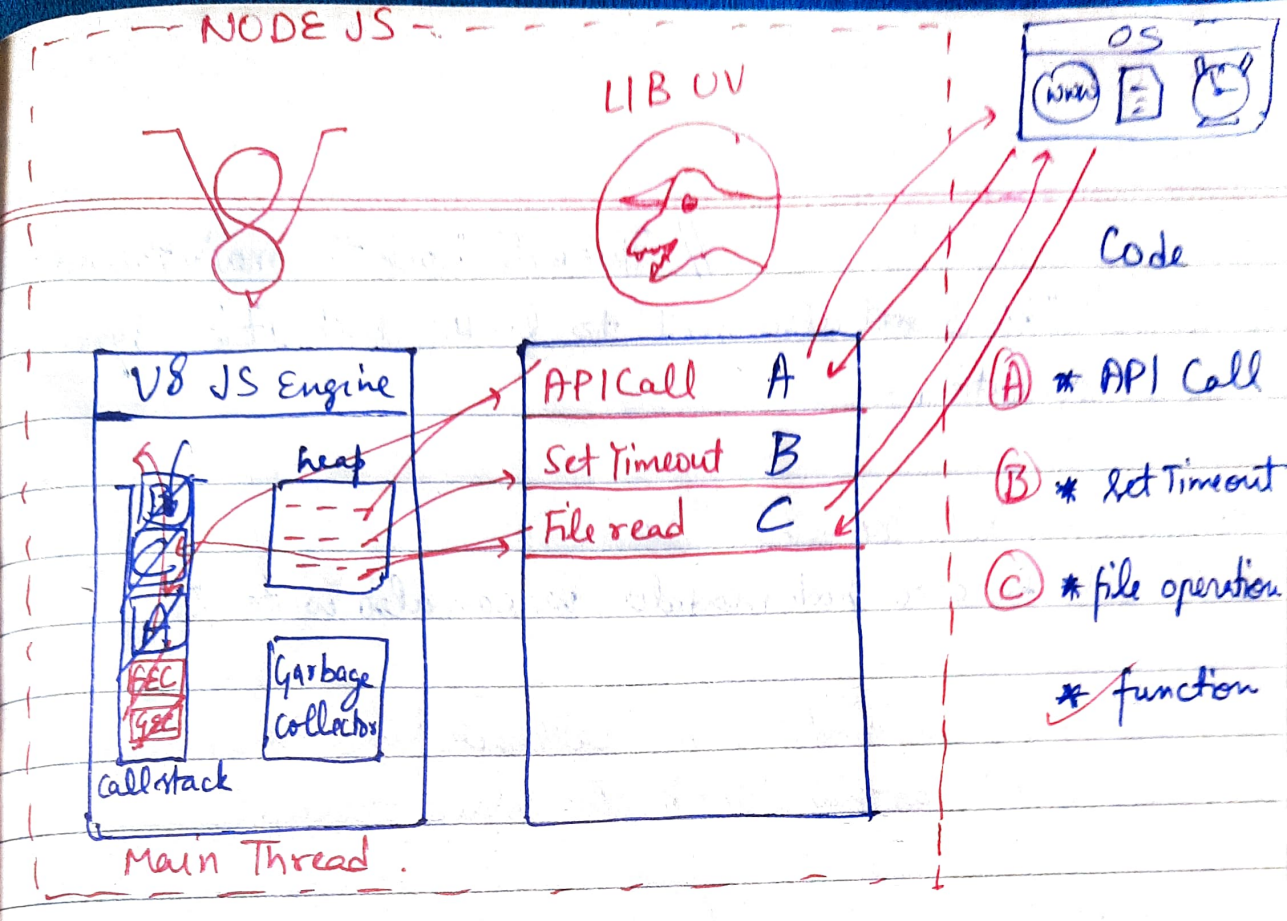
Conceptually, file accessing, connecting to DB and all needs IP address, permissions, so it is under OS only.




If any Async code comes to V8, it gives that to LIBUV & libuv acts as a medium b/w OS & V8 using NODEJS to get the work done.

LIBUV - Async IO made simple






* When function execution context is removed after returning the operation's value, then Garbage collector collects the var & clear the memory.

* libuv, interacts with OS, make api call or other things and then puts it in call stack to run the code.

# THIS IS HOW NODE JS IS CALLED ASYNC IO / Non-BLOCKING IO.

↓

we don't block the main thread for Async I/O.